



## C interfaces to GALAHAD FIT

Jari Fowkes and Nick Gould  
STFC Rutherford Appleton Laboratory  
Thu Jun 22 2023



---

<b>1 GALAHAD C package fit</b>	<b>1</b>
1.1 Introduction	1
1.1.1 Purpose	1
1.1.2 Authors	1
1.1.3 Originally released	1
<b>2 File Index</b>	<b>3</b>
2.1 File List	3
<b>3 File Documentation</b>	<b>5</b>
3.1 galahad_fit.h File Reference	5
3.1.1 Data Structure Documentation	5
3.1.1.1 struct fit_control_type	5
3.1.1.2 struct fit_inform_type	6
3.1.2 Function Documentation	6
3.1.2.1 fit_initialize()	6
3.1.2.2 fit_information()	6
3.1.2.3 fit_terminate()	7



# Chapter 1

## GALAHAD C package fit

### 1.1 Introduction

#### 1.1.1 Purpose

Fit polynomials to function and derivative data.

Currently, only the control and inform parameters are exposed; these are provided and used by other GALAHAD packages with C interfaces.

#### 1.1.2 Authors

N. I. M. Gould and D. P. Robinson, STFC-Rutherford Appleton Laboratory, England.

C interface, additionally J. Fowkes, STFC-Rutherford Appleton Laboratory.

Julia interface, additionally A. Montoison and D. Orban, Polytechnique Montréal.

#### 1.1.3 Originally released

March 2010, C interface January 2022.



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

<a href="#">galahad_fit.h</a> . . . . .	5
---	---



## Chapter 3

# File Documentation

### 3.1 galahad\_fit.h File Reference

```
#include <stdbool.h>
#include <stdint.h>
#include "galahad_precision.h"
#include "galahad_cfunctions.h"
```

#### Data Structures

- struct [fit\\_control\\_type](#)
- struct [fit\\_inform\\_type](#)

#### Functions

- void [fit\\_initialize](#) (void \*\*data, struct [fit\\_control\\_type](#) \*control, int \*status)
- void [fit\\_information](#) (void \*\*data, struct [fit\\_inform\\_type](#) \*inform, int \*status)
- void [fit\\_terminate](#) (void \*\*data, struct [fit\\_control\\_type](#) \*control, struct [fit\\_inform\\_type](#) \*inform)

#### 3.1.1 Data Structure Documentation

##### 3.1.1.1 struct fit\_control\_type

control derived type as a C struct

##### Data Fields

bool	f_indexing	use C or Fortran sparse matrix indexing
int	error	error and warning diagnostics occur on stream error
int	out	general output occurs on stream out
int	print_level	the level of output required is specified by print_level
bool	space_critical	if space_critical is true, every effort will be made to use as little space as possible. This may result in longer computation times
bool	deallocate_error_fatal	if deallocate_error_fatal is true, any array/pointer deallocation error will terminate execution. Otherwise, computation will continue
char	prefix[31]	all output lines will be prefixed by .prefix(2:LEN(TRIM(.prefix))-1) where .prefix contains the required string enclosed in quotes, e.g. "string" or 'string'

### 3.1.1.2 struct fit\_inform\_type

inform derived type as a C struct

#### Data Fields

int	status	return status. Possible values are: <ul style="list-style-type: none"> <li>• 0 Normal termination with the required fit.</li> <li>• -1. An allocation error occurred. A message indicating the offending array is written on unit control.error, and the returned allocation status and a string containing the name of the offending array are held in inform.alloc_status and inform.bad_alloc respectively.</li> <li>• -2. A deallocation error occurred. A message indicating the offending array is written on unit control.error and the returned allocation status and a string containing the name of the offending array are held in inform.alloc_status and inform.bad_alloc respectively.</li> <li>• -3. the restriction <math>n \geq 1</math> has been violated.</li> </ul>
int	alloc_status	the status of the last attempted allocation/deallocation.
char	bad_alloc[81]	the name of the array for which an allocation/deallocation error occurred.

## 3.1.2 Function Documentation

### 3.1.2.1 fit\_initialize()

```
void fit_initialize (
    void ** data,
    struct fit_control_type * control,
    int * status )
```

Set default control values and initialize private data

#### Parameters

in, out	<i>data</i>	holds private internal data
out	<i>control</i>	is a struct containing control information (see <a href="#">fit_control_type</a> )
out	<i>status</i>	is a scalar variable of type int, that gives the exit status from the package. Possible values are (currently): <ul style="list-style-type: none"> <li>• 0. The initialization was succesful.</li> </ul>

### 3.1.2.2 fit\_information()

```
void fit_information (
```

```
void ** data,
struct fit_inform_type * inform,
int * status )
```

Provides output information

#### Parameters

in, out	<i>data</i>	holds private internal data
out	<i>inform</i>	is a struct containing output information (see <a href="#">fit_inform_type</a> )
out	<i>status</i>	is a scalar variable of type int, that gives the exit status from the package. Possible values are (currently): <ul style="list-style-type: none"> <li>• 0. The values were recorded succesfully</li> </ul>

#### 3.1.2.3 fit\_terminate()

```
void fit_terminate (
    void ** data,
    struct fit_control_type * control,
    struct fit_inform_type * inform )
```

Deallocate all internal private storage

#### Parameters

in, out	<i>data</i>	holds private internal data
out	<i>control</i>	is a struct containing control information (see <a href="#">fit_control_type</a> )
out	<i>inform</i>	is a struct containing output information (see <a href="#">fit_inform_type</a> )

